Project Acronym: MEDIS

Project Title: A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems

Contract Number: 544490-TEMPUS-1-2013-1-ES-TEMPUS-JPCR

Starting date: 01/12/2013 **Ending date:** 30/11/2016

Deliverable Number: 2.3

Title of the Deliverable: AIISM teaching resources - Mobile and Cloud Computing Platforms
Task/WP related to the Deliverable: Development of the AIISM teaching resources - Mobile and Cloud Computing Platforms

Type (Internal or Restricted or Public): Internal

Author(s): Radu Dobrin and Sasikumar Punnekkat

Partner(s) Contributing:

Contractual Date of Delivery to the CEC: 30/09/2014

Actual Date of Delivery to the CEC: 30/09/2014

Project Co-ordinator

Company name :	Universitat Politecnica de Valencia (UPV)
Name of representative :	Houcine Hassan
Address :	Camino de Vera, s/n. 46022-Valencia (Spain)
Phone number :	+34 96 387 7578
Fax number :	+34 963877579
E-mail :	husein@upv.es
Project WEB site address :	https://www.medis-tempus.eu

Context

WP 2	Design of the AIISM-PBL methodology
WPLeader	Universitat Politècnica deValència (UPV)
Task 2.3	Development of the AIISM teaching resources - Mobile and Cloud Computing Platforms
Task Leader	MDU
Dependencies	UPV, MDU, TUSofia, USTUTT, UP

Author(s)	Radu Dobrin and Sasikumar Punnekkat
Contributor(s)	
Reviewers	

History

Version	**Date**	**Author**	**Comments**
0.1	01/03/2014	Radu Dobrin and Sasikumar Punnekkat	Initial draft
1.0	19/09/2014		Final version

Table of Contents

1 Executive summary

WP 2.3 details the learning materials of the Advanced Industrial Informatics Specialization Modules (AIISM) related to the Mobile and Cloud Computing Platforms.

The contents of this package follows the guidelines presented in the MDU's documentation of the WP 1 (Mobile and Cloud Computing Platforms)

- The PBL methodology was presented in WP 1.1
- The list of the module's chapters and the temporal scheduling in WP 1.2
- The required human and material resources in WP 1.3
- The evaluation in WP 1.4

The rest of the document is organized as follows: Section 2 introduces the course and the outlines. Section 3 details the lectures, divided in subsections for each chapter. Section 4 describes the labs. There is a special subsection for each chapter. Section 5 gives an overview to the seminars. Each seminar has its own subsection. Finally section 7 lists the bibliography and the references.

2 Introduction

This lecture will be on reliability of embedded systems and usage of fault tolerance and testing for dependable systems design. Three lectures will be conducted focusing on reliability, fault tolerance and testing of mobile applications.

At the end of the lecture, the students will be able to:

- Have an overall view of reliability and how it is related to security
- Perform a simple reliability analysis using the project specification suggest a design for fault tolerance
- Understand basic testing methodologies

3 Lectures

The lecture 1 in this chapter introduces reliability starting from the big picture of dependability as well as positions it with respect to security. It then introduces some terminology and presents some case studies where the compromises of reliability of critical systems lead to catastrophes. It then introduces three commonly used techniques to improve reliability namely: fault tolerance, testing and safety analysis. Fault tolerance and testing are introduced in detail in lectures 2 and 3 respectively. Lecture 1 of this chapter is accessible in file Lecture 11_Reliability-1.pptx.

Lecture 2 presents one of the ways improve reliability of computer systems, namely using fault tolerance. The lectures focuses on fault tolerance using redundancy. The most common types of redundancy viz. temporal redundancy and spatial redundancy are outlined. Additionally, the concept of N-version programming that is central to the concept of redundancy is explained. Finally, the challenges involved in improving reliability of mobile applications are outlined. Lecture 2 of this chapter is accessible in file Lecture 12_Reliability-2.pptx.

The last lecture in this chapter deals with software testing as a means to improve reliability. The students are introduced to different software development life cycle (SDLC) models. They are then given an overview of testing in the big picture of different SDLCs. Specific SDLC models that give importance to testing are outlined. Further they are introduced to different phases/types of testing, after which they are given an overview of a test automation framework. Finally, unit testing is introduced to the students with a short example on how this is done. This lecture of this chapter is accessible in file Lecture 13_Reliability-3.pptx.

4 Lab

This chapter consists of three labs aimed at strengthening the fundamentals of students in the area of reliability.

In the first lab, students modify the water tank controller application such that it simulates an erroneous behaviour e.g., by dropping messages randomly. They will collect data representing the behaviour of the water tank controller for some number of simulations.

In the second lab, the students will implement temporal and spatial redundancy and collect data representing the behaviour of the water tank controller application under both the strategies.

In the third lab, the students will test their mobile application using unit testing.

5 Seminar

In the first seminar, the students will read and discuss the following articles and write a short summary along with their reflections.

1. N. G. Leveson, “High-pressure steam engines and computer software,” in Proceedings of the 14th International Conference on Software Engineering, 1992, http://sunnyday.mit.edu/steam.pdf
2. What really happened on Mars? http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html
3. N. G. Leveson, The role of software in spacecraft accidents, http://sunnyday.mit.edu/papers/jsr.pdf

In the second seminar, the students will discuss their lab in the class:

- Which type of redundancy is more suitable for mobile applications?

- Does redundancy affect security? Discuss this with respect to what you have learned in the previous lectures.
- How can your design reliable networks for such applications?

Perform a short literature survey and discuss the findings

In the final seminar, the students will read and discuss the following, and write a short report along with their reflections:

1. On the Effectiveness of Test-first Approach to Programming Erdogmus, H, IEEE Transactions on Software Engineering, 2005, Available from: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=shwart&index=an&req=5763742&lang=en
2. An Analysis and Survey of the Development of Mutation Testing , Yue and Mark Harman, IEEE Transactions on Software Engineering, 2011, Available from: http://crest.cs.ucl.ac.uk/fileadmin/crest/sebasepaper/JiaH10.pdfDiscuss the article in the class.

6 Miniproject

In the first mini project, the students will submit a report summarizing the articles and the discussions during the seminar. The report should include their reflections on the reliability aspects of mobile applications that control embedded systems.

In the second mini project, the students will summarize the following articles, while also including their reflections about the same (this projects aims to complement the second lab discussed above):

1) Algirdas A. Avizienis, The methodology of N version programming, http://www.cse.cuhk.edu.hk/~lyu/book/sft/pdf/chap2.pdf
2) J.C Knight and N.G. Leveson, An experimental evaluation of the assumption of independence in multiversion programming, IEEE Transactions on Software Engineering, http://sunnyday.mit.edu/papers/nver-tse.pdf
3) A A. Avizienis, M R Lyu, and W Schutz, In Search of Effective Diversity: A Six-Language Study of Fault-Tolerant Control Software, http://ftp.cs.ucla.edu/tech-report/198_-reports/870060.pdf
4) J.C Knight and N.G. Leveson, A reply to the criticisms of the Knight & Leveson experiment, ACM SIGSOFT Software Engineering Notes, http://sunnyday.mit.edu/critics.pdf

As the final mini-project in this chapter, the students will form groups of two, and run the test cases that they developed earlier in the lab on each others' code and write a report summarizing the results while also discussing the results in the context of N-version programming

Did the two code fail on the same test case?
Is N-version programming effective?

7 References

[1] Basic concepts and taxonomy of dependable and secure computing, Avizienis, A. ; Laprie, J.-C. ; Randell, B. ; Landwehr, C., IEEE Transactions on Dependable and Secure Computing, 2004

[2] An Experimental Evaluation Of The Assumption Of Independence In Multi-Version Programming, J. C. Knight , N. G. Leveson, IEEE Transactions on Software Engineering, 1986

[3] N. G. Leveson, “High-pressure steam engines and computer software,” in Proceedings of the 14th International Conference on Software Engineering, 1992

[4] Algirdas A. Avizienis, The methodology of N version programming, http://www.cse.cuhk.edu.hk/~lyu/book/sft/pdf/chap2.pdf

[5] J.C Knight and N.G. Leveson, An experimental evaluation of the assumption of independence in multiversion programming, IEEE Transactions on Software Engineering, http://sunnyday.mit.edu/papers/nver-tse.pdf

[6] A A. Avizienis, M R Lyu, and W Schutz, In Search of Effective Diversity: A Six-Language Study of Fault-Tolerant Control Software, http://ftp.cs.ucla.edu/tech-report/198_-reports/870060.pdf

[7] J.C Knight and N.G. Leveson, A reply to the criticisms of the Knight & Leveson experiment, ACM SIGSOFT Software Engineering Notes, http://sunnyday.mit.edu/critics.pdf